



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/607,591	06/27/2003	Mark Ronald Plesko	3382-64706	5996
26119 7590 02/04/2008 KLARQUIST SPARKMAN LLP 121 S.W. SALMON STREET SUITE 1600 PORTLAND, OR 97204				
			EXAMINER PHAM, CHRYSTINE	
			ART UNIT 2192	PAPER NUMBER
			MAIL DATE 02/04/2008	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/607,591

Applicant(s)

PLESKO ET AL.

Examiner

Chrystine Pham

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 29 October 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,3-7,9,10,12-15,17-24 and 26-32 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1,3-7,9,10,12-15,17-24 and 26-32 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date 11/13/2007 & 12/21/2007.
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to Amendments filed on October 29, 2007. Claims 1 and 6 have been amended. Claim 8 has been canceled. Claims 1, 3-7, 9-10, 12-15, 17-24, 26-32 are presented for examination.

Response to Arguments

2. Applicant's arguments with respect to new claim limitation "wherein the one or more representations of the intermediate language are capable of representing programs written in a plurality of different source languages, wherein the plurality of different source languages comprise at least one typed source language and at least one untyped source language" (currently recited in claim 1) have been considered but are moot in view of the new ground(s) of rejection. See Gordon et al. (US 6,560,774 B1).
3. Other arguments filed October 19, 2007 have been fully considered but they are not persuasive.

Applicants essentially re-present the previous argument, "Knoblock does not describe rules for type checking a type designated (sic) as an unknown type" (Remarks, page 9 of 12, first ongoing paragraph). However, in the same paragraph, Applicants acknowledge "Knoblock determines constraints for array types so that Knoblock can assign types". Applicants also point out, "Knoblock

describes one method for **reconstructing types** involving **labeling variables as unknown**, collecting constraints between known types and unknown types, and **solving for the unknown types using the constraints**. Knoblock, col.8, lines 4-55" (Remarks, page 8 of 12, 1st full paragraph)(Emphasis added). Needless to say, the constraints anticipate the rules for resolving unknown types.

Furthermore, since Knoblock is directed to a method of type-checking the intermediate code (see at least col.5:30-col.6:14), which comprises type information that is lost during the translation from source code to intermediate code, it is why Knoblock collects constraints and resolve the unknown types in the intermediate code, i.e., to type-check (i.e., verify) the intermediate code.

Applicants further argue, "Franz does not teach or suggest that a type designated as an **unknown type** with associated machine-representation size information associated with the unknown type" (Remarks, page 11 of 12, 1st ongoing paragraph)(Emphasis added). However, it is respectfully submitted that Applicants' argument is based on a piecemeal analysis of the Franz reference because Franz has not been relied upon to suggest the "unknown type". Rather, as has been established in the previous Office Action and acknowledged by Applicants (Remarks, page 9 of 12, ongoing paragraph), Knoblock clearly teaches collecting constraints for **unknown array types** (Emphasis added). Since, as established in the previous Office Action (page 9), both Knoblock and Franz are directed to type checking. Furthermore, as Applicants acknowledge, "Franz describes the need for enforcing **array index checking**"

(Remarks, page 11 of 12, ongoing paragraph) (Emphasis added) and "The **size of an array** may not be known statically, but once the array object has been created, its size will remain constant" (Remarks, page 10 of 12, last paragraph)(Emphasis added), it is clear that all arrays (e.g., unknown array types as disclosed by Knoblock) must have size associated with them. It is further inherent that the size of the array has a machine-representation in order for the machine to store the array, i.e., allocate the memory space corresponding to the size of the array. Thus, Knoblock as modified by Franz clearly discloses "a type designated as an unknown type with associated machine-representation size information associated with the unknown type".

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.
5. Claims 1, 3-7, 14, 17-23, 29-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Knoblock et al. of record (US 6981249 B1, "Knoblock") in view of Gordon et al. (US 6,560,774 B1, "Gordon").

Claim 1

Knoblock teaches a method of type-checking a code segment written in a programming language (see at least *type checking, compiler, interpreter* col.5:30-col.6:32)

comprising:

- translating the code segment from the programming language to one or more representations of an intermediate language (see at least *intermediate program* col.1:63-col.2:52; *translator 204, source program 202, bytecode program 206, intermediate program 210* col.5:30-col.6:32; 402 FIG.4 & associated text); and
- type-checking the one or more representations based on a rule set (see at least FIG.12 & associated text; *constraints, type reconstruction* col.14:4-24), wherein the rule set comprises rules for type-checking a type designated as the unknown type (see at least *bytecode program 206 lacks some of the types, reconstruct, type inference, type elaboration* col.5:55-col.6:2), wherein the unknown type indicates that an element of the representation is of a type that is not known (see at least *unknown type, intermediate program* col.1:63-col.2:52; *type reconstruction, type variable, unknown type* col.8:4-19; FIG.4 & associated text; col.13:35-52; col.7:20-45; col.8:19-30).

Knoblock does not expressly disclose wherein the one or more representations of the intermediate language are capable of representing programs written in a plurality of different source languages, wherein the plurality of different source languages comprise

at least one typed source language and at least one untyped source language.

However, Gordon discloses a system (see at least FIG.1 & associated text) and method for type-checking one or more representations of an intermediate language (see at least 202 FIG.2 & associated text), wherein the one or more representations of the intermediate language are capable of representing programs written in a plurality of different source languages (see at least 204 FIG.2 & associated text), wherein the plurality of different source languages comprise at least one typed source language (see at least *Java*, *C++*, *Visual Basic* col.35:8-20) and at least one untyped source language (see at least *Lips*, *Scheme*, *Smalltalk* col.35:8-20). Knoblock and Gordon are analogous art because they are both directed to type-checking. It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of Gordon into that of Knoblock for the inclusion of one or more representations of the intermediate language are capable of representing programs written in a plurality of different source languages, wherein the plurality of different source languages comprise at least one typed source language and at least one untyped source language. And the motivation for doing so would have been to enforce type safe code for multiple source languages (i.e., COM+) while maintaining optimal speed for the execution engine (see at least Gordon col.1:10-41).

Claim 3

The rejection of base claim 1 is incorporated. Knoblock further teaches wherein the rule set is selected from a plurality of rule sets (see at least *minimal solution*, *set of solutions*

col.14:20-24; FIG.12 & associated text).

Claim 4

The rejection of base claim 3 is incorporated. Knoblock further teaches wherein only a fraction of the plurality of rule sets contain rules for type-checking a type designated as the unknown type, wherein the unknown type indicates that an element of the representation is of a type that is not known (see at least *data member constraint 1202*, *data member 1204*, *unknown type* col.14:13-15).

Claim 5

The rejection of base claim 1 is incorporated. Knoblock further teaches wherein the rule set further comprises rules for type-checking types representing categories of types found in a plurality of programming languages (see at least *data member constraint 1202*, *data member 1204*, *unknown type*, *known type* col.14:13-15; FIG.12 & associated text);

Claim 6

Knoblock teaches a method of selectively retaining type information during compilation in a code segment written in a programming language (see at least *constructing 412* FIG.4 & associated text), the method comprising:
translating the code segment from the programming language to one or more representations of an intermediate language (see at least 402 FIG.4 & associated text);

for each representation, determining whether to retain type information for one or more elements of the representation; and based on the determination, associating one or more elements of the representation with a type, designated as the unknown type, indicating the element can be of any type (see at least *type reconstruction, type variable, unknown type, solution, new type* col.8:4-55; FIG.4 & associated text; col.13:35-52); and type-checking the one or more representations based on a rule set, wherein the rule set comprises rules for type-checking the type designated as the unknown type (see at least *data member constraint 1202, data member 1204, unknown type* col.14:13-15).

Claim 7

The rejection of base claim 6 is incorporated. Knoblock further teaches wherein the determination is based on a current stage of compilation, a characteristic of each representation, or the programming language (see at least *type reconstruction, type variable, unknown type, solution, new type* col.8:4-55; FIG.4 & associated text).

Claims 14, 17-23

Claims recite limitations, which have been addressed in claims 1, 3-7, 9 and 10, therefore, are rejected for the same reasons as cited in claims 1, 3-7, 9 and 10.

Claim 29

Knoblock teaches a computer-readable medium containing computer-executable instructions for implementing the method of claim 1 (see at least FIGS.2A-2B & associated text).

Claim 30

The rejection of base claim 1 is incorporated. Knoblock further teaches wherein the rule set further comprises rules for dropping type information for one or more elements of the representation by changing a known type of the one or more elements to the type designated as the unknown type (see at least FIG.3B & associated text; *type, local variable, constraint collection, type reconstruction, type variable, unknown type* col.7:20-col.8:55).

Claim 31

The rejection of base claim 6 is incorporated. Claim recites limitations, which have been addressed in claim 30, therefore, is rejected for the same reasons as cited in claim 30.

6. Claims 9, 10, 12-13, 15, 24, 26-28 and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Knoblock in view of Gordon further in view Franz et al. of record (US 7117488 B1, "Franz").

Claim 9

The rejection of base claim 6 is incorporated. Knoblock further teaches wherein the type, designated as the unknown type, indicating the element can be of any type (e.g., array)(see at least *type variable, unknown type, integer types, Booleans, bytes, shorts* col.8:4-47; *unknown array types* col.13:35-40). Knoblock does not expressly disclose said element (i.e., array) has size information associated with it. However, Franz teaches indicating an element can be of type array and has size information associated with it (see at least *array, size* col.11:63-col.12:11). Knoblock and Franz are analogous art because they are both directed to type checking. It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of Franz into that of Knoblock for the inclusion of size information associated with array types. And the motivation for doing so would have been enforce type safe code (see at least Franz col.1:55-col.2:55).

Claim 10

The rejection of base claim 9 is incorporated. Knoblock further disclose generating code from at least elements associated with the type, designated as the unknown type indicating the element can be of any type)(see at least *type variable, unknown type, integer types, Booleans, bytes, shorts* col.8:4-47; *unknown array types* col.13:35-40). Knoblock does not expressly disclose said indicating is based on the size information. However, indicating the element can be of any type based on the size information of a machine representation (see at least *array, size* col.11:63-col.12:11).

Claim 12

Knoblock teaches a method of translating types associated with a plurality of programming languages to types of an intermediate language (see at least FIG.4 & associated text; *type reconstruction, type variable, local variable* col.8:4-55), the method comprising:

replacing the types associated with the plurality of programming languages with the types of the intermediate language, wherein the types of the intermediate language comprise general categories of the types associated with the plurality of programming languages and a type designated as an unknown type (see at least 502, 504 FIG.5 & associated text), wherein the type designated as the unknown type has size information associated with it, wherein the size information comprises size information of a machine representation of the type designated as the unknown type (see at least col.11:63-col.12:11).

Claim 13

The rejection of base claim 12 is incorporated. Knoblock further teaches wherein the types of the intermediate language further comprise types related to programming language specific primitive types (see at least 414 FIG.4 & associated text; col.8:43-55).

Claims 15 and 32

Claims recite limitations, which have been addressed in claim 10, therefore, therefore, are rejected for the same reasons as cited in claim 10.

Claims 24, 26-27

Claims recite limitations, which have been addressed in claims 1, 3-7, 9 and 10, therefore, are rejected for the same reasons as cited in claims 1, 3-7, 9 and 10.

Claim 28

Knoblock teaches a computer-readable medium containing computer-executable instructions for implementing the method of claim 24 (see at least FIGS.2A-2B & associated text).

Conclusion

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chrystine Pham whose telephone number is 571-272-3702. The examiner can normally be reached on Mon-Fri, 8:30am-5pm.


If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through

Application/Control Number:
10/607,591
Art Unit: 2192

Page 13

Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.



TUANDAM
SUPERVISORY PATENT EXAMINER